



Monde professionnel de la Science des Données

Structuration des données M2 Data Science

Présentation Sommaire

Lucas HO

Avril 2023 - Data Scientist (~2 ans) chez ALTEN:
Cabinet de conseil dans l'ingénierie

Sous-traitance Airbus Helicopters :
Product Manager
Dev & Data Full Stack
Data Scientist

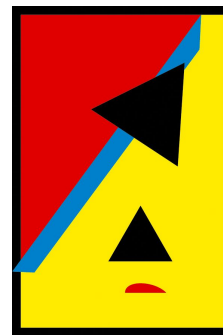
Problématiques Export Control

Ancien élève du Master Data Science (2020-2022)

Contact :

lucas.ho@alten.com

lucas.ho@univ-amu.fr



ALTEN



AIRBUS
HELICOPTERS



Organisation

3 CM de 3h:

4.5h Cours + 4.5h TP

Contenu:

- Comment créer une database
- Initiation à un outils de database
- Requêtes / SQL

⇒ 1 TD final de 2h

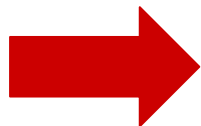
Data Scientist - Les attendus

Allons voir les annonces de poste Data Scientist sur Internet

Analyse Rapide des propositions de postes récentes:

1. R/Python
2. ML/DL
3. SQL/NoSQL...
4. GCP/Azure/AWS

Soft Skills aussi

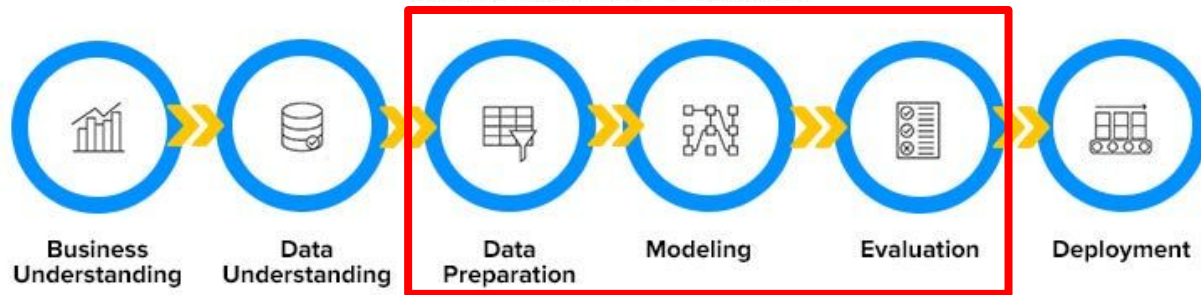


Nécessaire de comprendre une database / faire des requêtes

Les étapes en Data Science

Focus fait au cours du M2

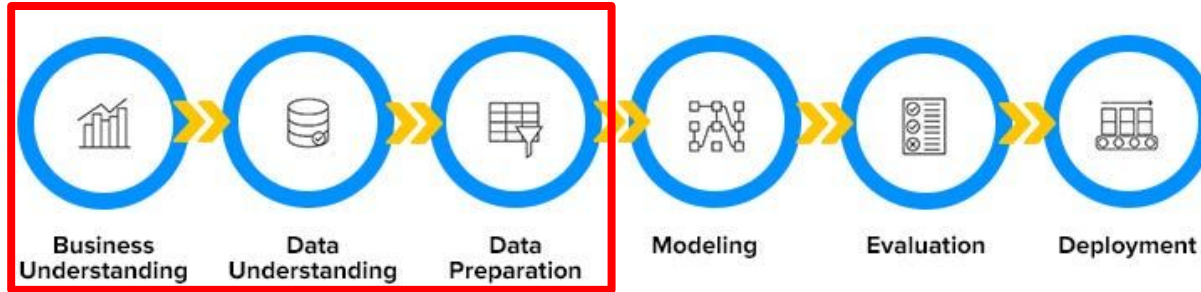
Data Science Process



Les étapes en Data Science

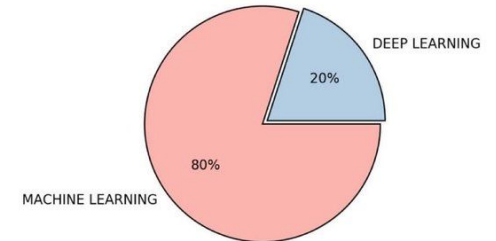
Ce qu'il se passe dans le métier

Data Science Process



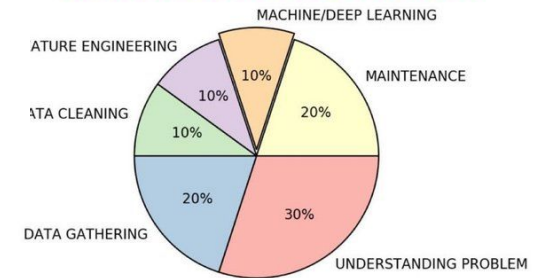
DATA SCIENTIST JOB - EXPECTATION

@drangshu



Follow: Dr. Angshuman Ghosh

DATA SCIENTIST JOB - REALITY



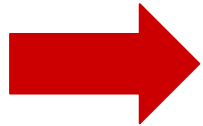
bit.ly/drangshu

L'environnement Data

Dans quel environnement un Data Scientist travaille / avec quoi il travaille ?

Idéalement : Data Lake / Data Warehouse

Ex : Airbus Helicopters



Nécessité de mettre en place une database

Data Lake vs Data Warehouse

Data Lake est un vaste gisement (pool) de données brutes dont le but n'a pas été précisé.

Data Warehouse est un référentiel de données structurées et filtrées qui ont déjà été transformées dans un but spécifique.

Données Structurées vs Données Non structurées

Données structurées

Très précises et stockées dans un format prédéfini

Données non structurées

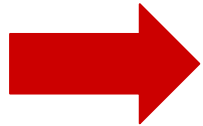
Conglomération de nombreuses données de différents types qui sont stockées dans leurs formats en mode natif.

L'environnement Data

Dans quel environnement un Data Scientist travaille / avec quoi il travaille ?

Idéalement : Data Lake / Data Warehouse

Ex : Airbus Helicopters



Nécessité de mettre en place une database

Database?

Mais comment structurer faire une database?

- **De quoi j'ai besoin pour la créer?**
- **Sur quelle interface?**
- **Qu'est-ce qui compose ma database?**
- **Pour qui cela va être utile?**
- ...



Contenu

I-Création d'une database

II-Exploitation/Interrogation d'une database

III- Entretien d'une database

TP Final

The background of the slide is an aerial photograph of a dry, cracked landscape, possibly a desert or a dried-up lake bed. The image is overlaid with a semi-transparent blue filter. In the center, there is a white rectangular box containing the text 'I - Création d'une database' in a bold, dark blue font.

I - Création d'une database

Création d'une database?

Par où commencer ?

Solution : Mettre à disposition les données directement et on verra plus tard?

*Carpe
Diem*



Data Modeling - C'est quoi?

Prenons pour exemple ces deux datasets:

POLICY TABLE

Policy Number	Date Issued	Policy Type	Customer Number	Commission Rate	Maturity Date
V213748	02/29/1989	E20	HAYES01	12%	02/29/2009
N065987	04/04/1984	E20	WALSH01	12%	04/04/2004
W345798	12/18/1987	WOL	ODEAJ13	8%	06/12/2047
W678649	09/12/1967	WOL	RICHB76	8%	09/12/2006
V986377	11/07/1977	SUI	RICHB76	14%	09/12/2006

CUSTOMER TABLE

Customer Number	Name	Address	Postal Code	Gender	Age	Birth Date
HAYES01	S Hayes	3/1 Collins St	3000	F	25	06/23/1975
WALSH01	H Walsh	2 Allen Road	3065	M	53	04/16/1947
ODEAJ13	J O'Dea	69 Black Street	3145	M	33	06/12/1967
RICHB76	B Rich	181 Kemp Rd	3507	M	59	09/12/1941

Questions:

Customer Number - Policy= Customer

Number - Customer ?

Possibilité de regrouper en une seule table?

Pourquoi une colonne Age et une colonne Birth Date?

Policy type et Commission rate équivalent?

But du Data Modeling ?

⇒ Résoudre/Éviter ces questions/problèmes

Data Modeling - C'est quoi?

Définition

Processus de création d'un modèle de données pour représenter visuellement la structure des données via un diagramme, les relations et les flux entre différents éléments dans un système d'information.

L'accent est alors mis sur le besoin de disponibilité et d'organisation des données.

<https://aws.amazon.com/fr/what-is/data-modeling/>

https://atlan-com.translate.goog/what-is-data-modeling/?_x_tr_sl=en&_x_tr_tl=fr&_x_tr_hl=fr&_x_tr_pto=sc

<https://www.simplilearn.com/what-is-data-modeling-article>

Data Modeling - C'est quoi?

Intérêts

- Permet aux entreprises de transformer des ensembles de données brutes en informations exploitables, facilitant ainsi la prise de décision, la gestion des ressources et l'amélioration des processus commerciaux.
- Cela permet d'améliorer l'analyse des données
- Fournit le schéma directeur pour la construction d'une nouvelle base de données ou la réorganisation d'applications existantes.
- Améliore la cohérence du nommage, des règles, de la sémantique et de la sécurité.
- Méthode standardisée pour définir et mettre en forme les contenus de la base de données de manière cohérente dans tous les systèmes, permettant à différentes applications de partager les mêmes données.
- Structure des relations entre les éléments de données au sein d'une base de données, ainsi qu'un guide d'utilisation des données.

...

Data Modeling - C'est quoi?

Intérêts

Grosso modo:

- Data organization
- Data clarity
- Data integration
- Data quality assurance
- Efficient querying
- Scalability
- Decision support
- Communication
- Database design
- Data governance
- Change management
- Documentation

Gérer une database - Différents outils

Différents outils pour gérer une database



Pour ce cours GCP/AWS ⇒ :'(

Pourquoi PostgreSQL : <https://survey.stackoverflow.co/2022/#most-popular-technologies-database>

Installation d'un environnement pour gérer une database

Avant de continuer le cours,

Installation de PostgreSQL pour plus tard...

Data Modeling - Différents Niveaux d'abstractions

1. **Conceptual data model**
2. **Logical data model**
3. **Physical data model**

Data Modeling - Différents Niveaux d'abstractions

1. Conceptual data model

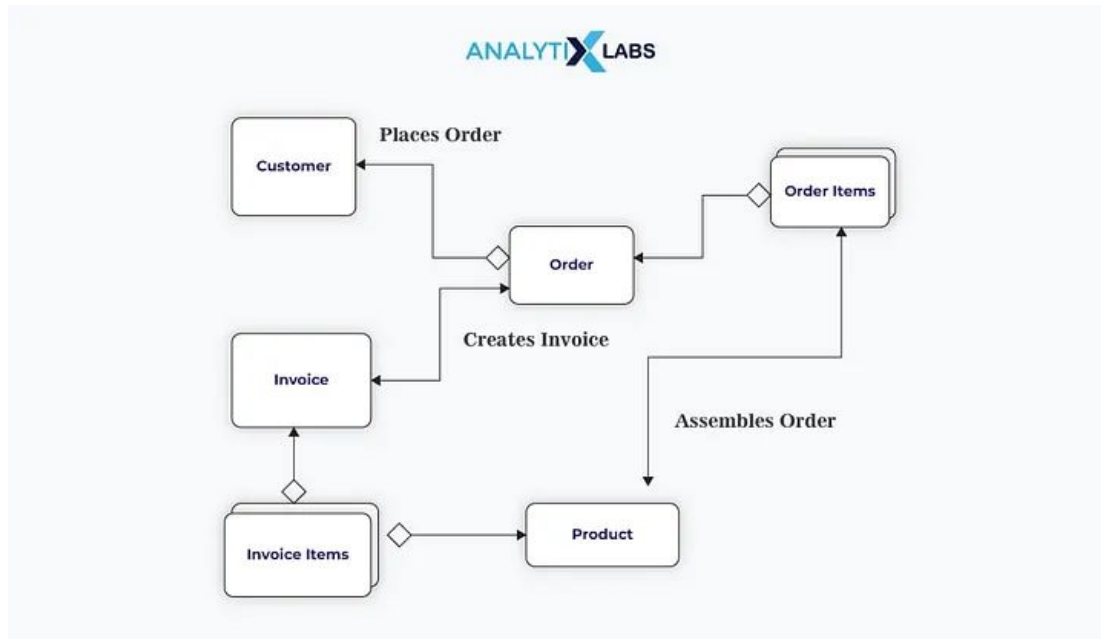
Aussi appelé domain model

Donne une vue générale de que le système contient, comment il va être organisé, quelles règles Business le défini....

Généralement créé pour déterminer les exigences initiales

On va avoir:

- Entités
- Caractéristiques / Contraintes
- Inter-relation
- Exigence en terme de Sécurité/ Intégrité



Data Modeling - Différents Niveaux d'abstractions

2. Logical data model

Moins abstrait et détaille plus les concepts / relations

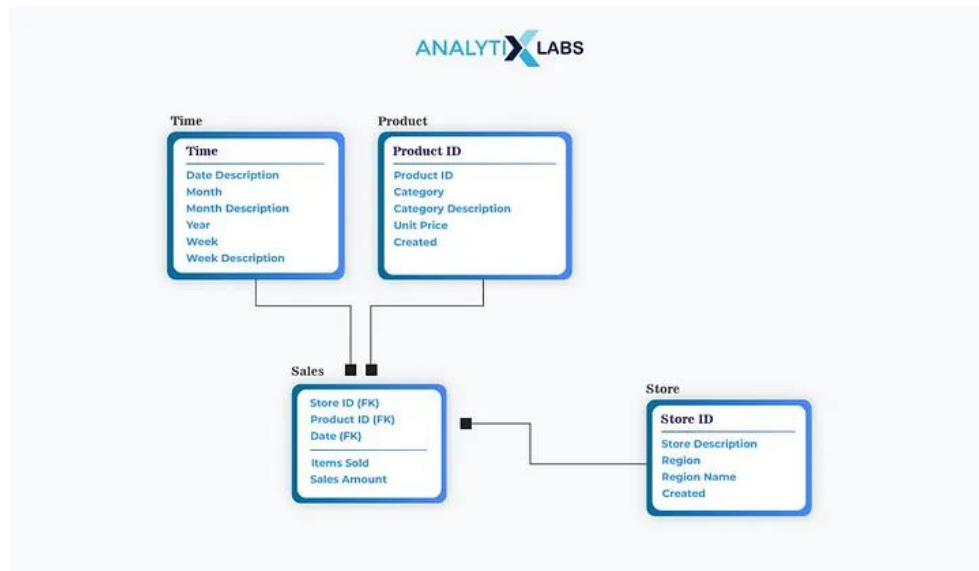
Indique:

- Attribut des données / Variables
- Relation entre les entités plus en détail

Les modèles de données logiques ne spécifient aucune exigence technique du système ⇒ Pas directement utilisable

Étape souvent omise dans les pratiques agiles / DevOps

Utile dans des environnements de mise en œuvre très procéduraux, ou pour des projets orientés données par nature (Data Warehouse)



Clé Primaire vs Clé Étrangère

Clé primaires vs Clé étrangères

Clé primaire (PK): Colonne qui identifie un enregistrement dans une table. Cette dernière doit être unique et ne peut être nulle. Il ne peut y avoir qu'une seule PK par table

Pas obligatoire mais vivement conseillée

id	nom	Prénom	age	ville
1	Dujardin	Marion	33	Nîmes
2	Tautou	Jean	52	Bordeaux
3	Richard	Audrey	26	Lille
4	Cotillard	Pierre	41	Strasbourg

Clé Primaire vs Clé Étrangère

Clé primaires vs Clé étrangères

Clé étrangère (FK): Colonne dans laquelle les valeurs correspondent à la colonne d'une clé primaire appartenant à une autre table. Elle est utilisée pour relier 2 tables ensemble

Table avec la PK : Référence/Table Parent
Table avec la FK : Table enfant

Une table peut contenir plusieurs clés étrangères

Animaux				
id	animal	nom	age	proprietaire_id
1	Chat	Moustache	3	3
2	Chien	Snoopy	5	1
3	Lapin	Carotte	2	1
4	Poisson	Bubulle	1	2

Data Modeling - Différents Niveaux d'abstractions

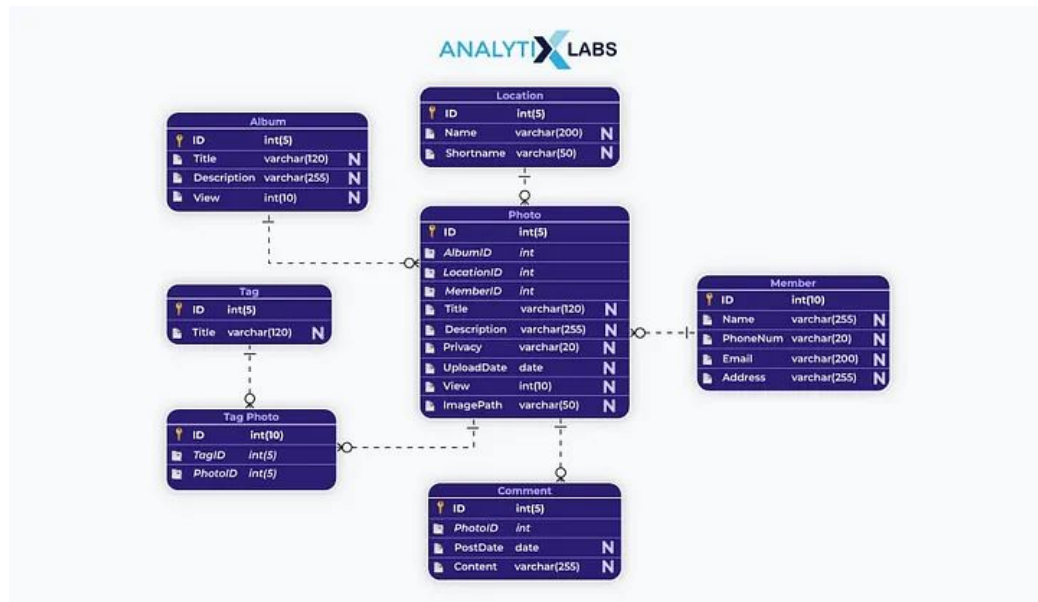
3. Physical data model

Schéma de comment les données vont être physiquement stockées dans la database

Abstraction la plus détaillée

Design final qui peut être implémentée dans une database

- Relation entre les entités
- primary keys
- foreign keys
- Type des variables



Recap: conceptual vs. logical vs. physical data model

1. Conceptual data model:

- Se concentre sur des concepts et des relations de haut niveau
- Représente les besoins de l'entreprise sans détails techniques
- Vise à parvenir à une compréhension commune entre les parties prenantes

2. Logical data model:

- Ajoute plus de détails au modèle conceptuel
- Définit les entités, les attributs, les relations et les clés.
- Représentation des éléments de données indépendante de la technologie

3. Physical data model:

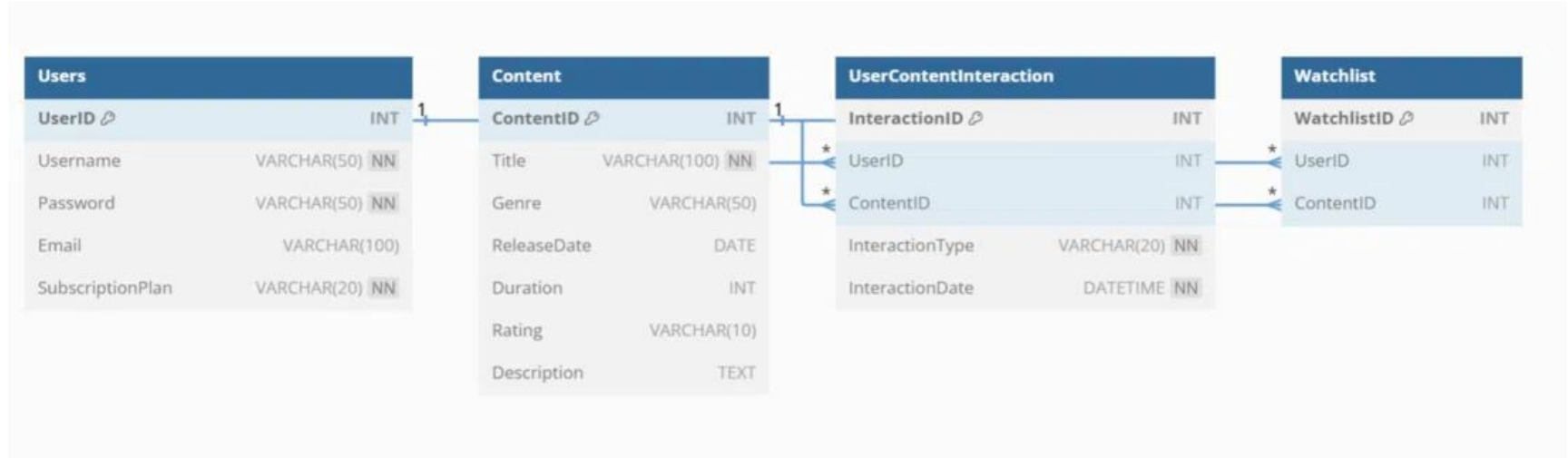
- Traduit le modèle logique en spécifications techniques
- Spécifie les types de données, les contraintes et les détails spécifiques à la base de données
- Prêt à être mis en œuvre dans un système de base de données particulier

Data Modeling - Autres Types

- Hierarchical
- Entity-Relationship
- Object-Oriented
- Network
- Relational (le plus récurrent et celui que l'on voit)

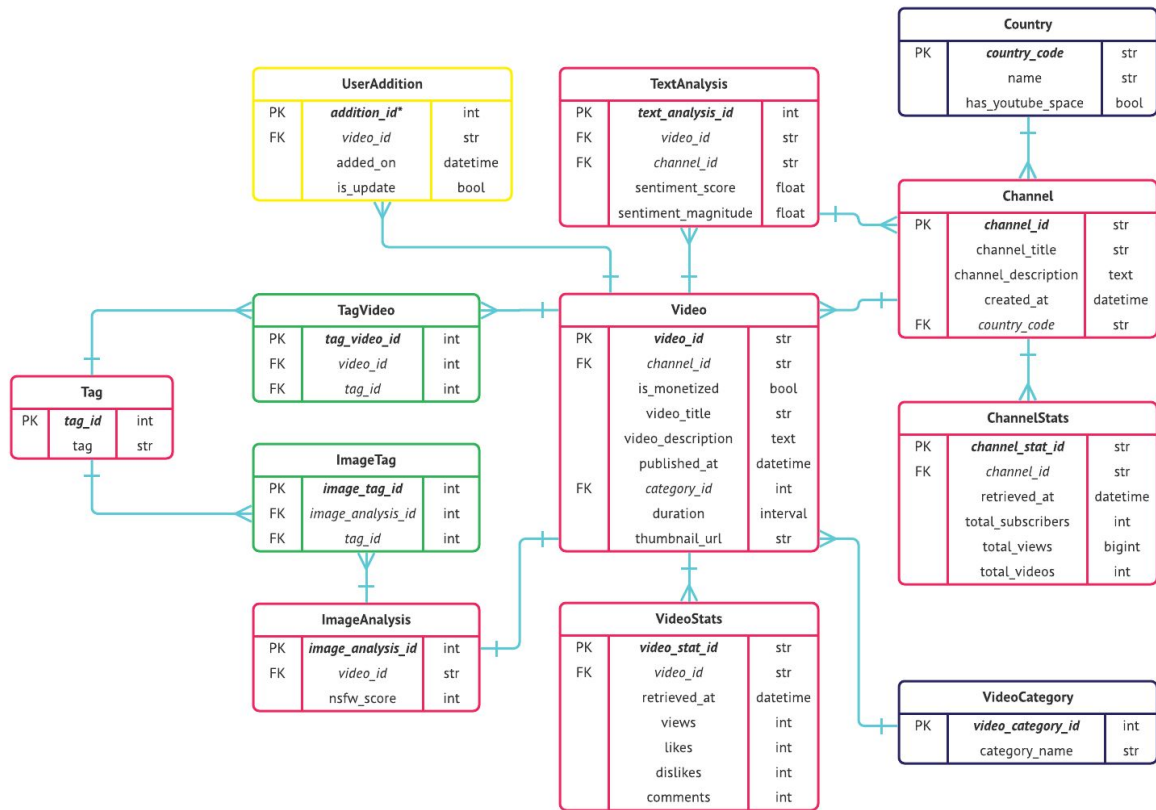
Data Model - Examples (1)

Netflix



Data Model - Examples (2)

Youtube



Exercice Pratique - CM1 Part 1

Faites des groupes de 3-4 personnes:

Mettez en place un data Modeling pour les cas suivants:

Type de database:

- Accords de Vente de pièces hélico
- [Réseau social](#)
- Outil de service RH
- Service de vidéo en ligne
- Ecommerce

Data Modeling Idéal ?

Malheureusement, pas de Data Model Final

Comment faire du Data Modeling?

1. Requirements analysis
2. Conceptual modeling
3. Logical modeling
4. Physical modeling
5. Maintenance and optimization

Data Modeling - Pièges

- **Sur-modélisation** : Ne créez pas un modèle trop complexe. La simplicité est souvent la clé.
- **Ignorer l'évolutivité** : Votre modèle doit pouvoir s'adapter à la croissance de votre entreprise.
- **Négliger la sécurité** : Intégrez les considérations de sécurité dès le début de votre modélisation.
- **Oublier la documentation** : Un modèle non documenté est un modèle inutilisable à long terme.

Tendance passagère

Investissement pour plus tard

Préparation pour les problématiques futurs

Data Modeling - Mistakes

Erreurs classiques à éviter:

- Prendre un champ lié au Business en PK
- Stocker des données redondantes
- Peu de Data Integrity / Peu de contrainte (PK, FK, UNIQUE, NOT NULL, CHECK)
⇒ Duplicates/Id inexistants
- Avoir plusieurs informations dans un champ
- Stocker des champ optionnel dans des colonnes différentes
- Choisir le mauvais data type / taille

Data Modeling - Outils

Différents outils possible pour faire du Data Modeling:

- MySQL Workbench
- ER/Studio
- Erwin Data Modeler
- Draw.io
- Squirrel SQL Client
- dbdiagram.io

Data Model Types - Schéma en étoile

Approche de modélisation mature largement adoptée par les data warehouses, les bases de données, les data marts.

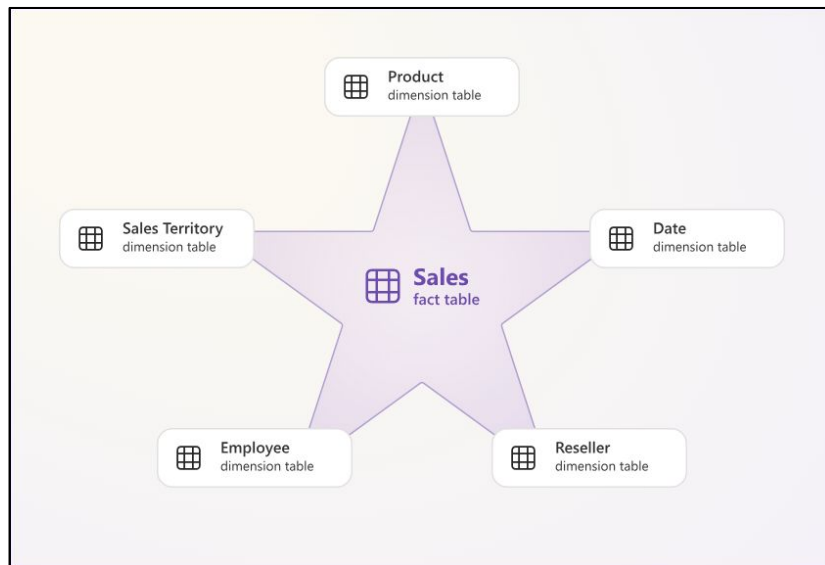
Les **tables de dimension** décrivent les entités d'entreprise : ce que l'on modélise

Ex: produits, personnes, lieux, temps ...

La table la plus cohérente que vous trouverez dans un schéma en étoile est une table de dimension de date. Une table de dimension contient une colonne clé (ou des colonnes) qui agit comme un identificateur unique et d'autres colonnes.

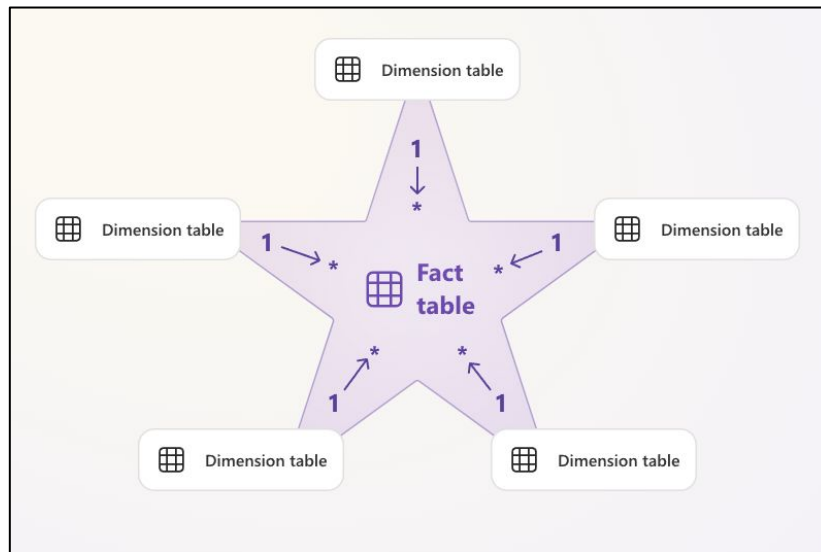
Les **tables de faits** stockent des observations/événements

Ex: commandes commerciales, des soldes boursiers, des taux de change, des températures...



Data Model Types - Schéma en étoile

- Optimisés pour interroger de grands ensembles de données.
- Efficaces pour stocker/ maj des données, tout en conservant un historique fiable
- réduisent la duplication de définitions métier répétitives et accélèrent l'agrégation et le filtrage des données




Data Model Types - Schéma en étoile

Normalisation et dénormalisation


Normalisation : données stockées de manière à réduire les données répétitives.

Considérez une table de produits qui a une colonne de valeur clé unique, comme la clé de produit et d'autres colonnes qui décrivent les caractéristiques du produit, telles que le nom du produit, la catégorie, la couleur et la taille. Une table de ventes est considérée normalisée quand elle stocke uniquement des clés, comme la clé de produit. Dans l'image suivante, notez que seule la ProductKey colonne enregistre le produit.

 Normalized table

OrderNumber	OrderDate	ProductKey	ResellerKey	SalesAmount
SO69561	2024-05-04	594	546	226.00
SO69560	2024-05-04	513	100	218.45
SO69560	2024-05-04	594	100	113.00
SO69539	2024-04-31	243	529	858.90
SO69539	2024-04-31	378	529	1146.01



 Denormalized table

OrderNumber	OrderDate	ProductKey	Product	Category	Color	Size	ResellerKey	SalesAmount
SO69561	2024-05-04	594	Mountain-500 Silver, 48	Bikes	Silver	48	546	226.00
SO69560	2024-05-04	513	ML Mountain Frame-W - Silver, 46	Components	Silver	46	100	218.45
SO69560	2024-05-04	594	Mountain-500 Silver, 48	Bikes	Silver	48	100	113.00
SO69539	2024-04-31	243	HL Road Frame - Red, 44	Components	Red	44	529	858.90
SO69539	2024-04-31	378	Road 250 - Black, 52	Bikes	Black	52	529	1146.01

Data Model Types - Schéma en étoile

Avantages	Inconvénient
<ul style="list-style-type: none">• Interrogation simplifiée : Leur structure dénormalisée réduit le nombre de jointures nécessaires pour récupérer les données. Cela simplifie et conduit à une agrégation et à des rapports de données plus rapides.• Performances plus rapides : La complexité réduite des jointures et l'indexation ⇒ améliorent la récupération des données ⇒ accès rapide aux informations.• Analyse intuitive : Les schémas en étoile permettent une analyse de données intuitive et simple. Les utilisateurs peuvent facilement comprendre les relations et les hiérarchies entre les dimensions.• Prise en charge robuste : Les schémas en étoile prennent en charge les structures OLAP	<ul style="list-style-type: none">• Manque d'intégrité : La dénormalisation peut entraîner une redondance des données ⇒ problèmes de qualité des données. Des modifications fréquentes peuvent également entraîner l'affichage d'informations obsolètes dans certaines tables.• Coûts accrus : L'ajout de données redondantes augmente les coûts de calcul et de stockage• Flexibilité limitée : construits pour des cas d'utilisation spécifiques. D'autres approches pourraient être plus efficaces pour les requêtes complexes impliquant plusieurs jointures.

Data Model Types - Schéma en étoile

Meilleure option lorsque :

- Les utilisateurs ont une compréhension claire des données requises
- Les données sont structurées et quantitatives avec quelques attributs catégoriels
- Ils veulent les données rapidement et facilement, sans créer de multiples jointures ⇒ La performance des requêtes est la priorité absolue
- La redondance des données ne sera pas un problème

Exercice Pratique - CM1 Part 2

Suite au point avec le Business, vous avez réussi à définir un Data model viable.

Il est maintenant temps de mettre en place ce Data Model...

Cf. voir fiche exercice

The background of the slide is an aerial photograph of a dry, cracked landscape, possibly a desert or a dried-up lake bed. The image is overlaid with a semi-transparent blue filter. In the center, there is a white rectangular box containing the title text.

II - Exploitation/Interrogation d'une database

Interroger la database - SQL

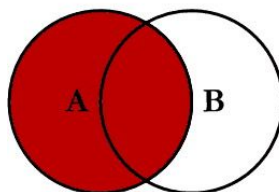
Types de données

Data Type	Description	Ex
INT	Tous les nombres	Âge, quantité,
NUMERIC(P,S)	Nombre décimaux	Taille, Prix, ...
SERIAL	auto-incrémentation	id(clé primaire)
CHAR(N)	Nombre de caractères fixe (Longueur fixe N)	Civilité, pays, ...
VARCHAR(N)	Nombre de caractères varié (Longueur max N)	Nom, Email, ...
TEXT	Nombre de caractères illimité	Commentaire, Description,...
TIME	HH:MM:SS	Heure du post
DATE	AAAA-MM-JJ	Date de naissance
TIMESTAMP	AAAA-MM-JJ HH:MM:SS	Date et heure de commande
BOOLEAN	VRAIX/FAUX	En Stock ?
ENUM	Liste de valeurs à sélectionner par l'utilisateur	Jour de la semaine

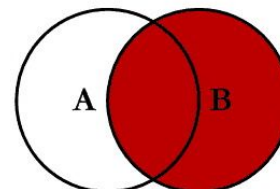
Transformation de données - Jointures

Focus sur les jointures:

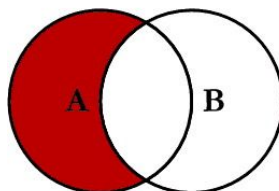
SQL JOINS



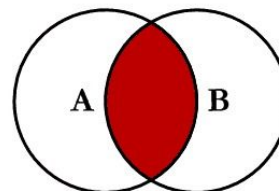
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



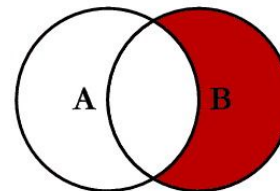
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



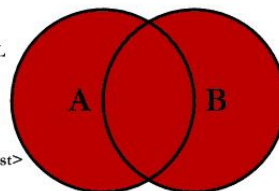
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



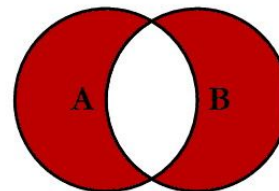
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

Exercice Pratique - CM2

Cf. voir fiche exercice

SQL - Training

Leetcode:

<https://leetcode.com/studyplan/top-sql-50/>

<https://leetcode.com/problemset/database/?search=sql>

<https://leetcode.com/discuss/general-discussion/1208129/list-of-free-leetcode-sql-questions>

W3Schools:

<https://www.w3schools.com/sql/default.asp>

SQLZoo:

https://sqlzoo.net/wiki/SQL_Tutorial

do1thub:

<https://www.dolthub.com/discover>

HackerRank:

<https://www.hackerrank.com/domains/sql>

SQL - View vs Materialized View

Feature	View	Materialized View
Definition	A view is a virtual table created from a query, and it doesn't store data physically.	A materialized view stores the results of a query physically in the database for faster retrieval.
Data Storage	Only the query expression is stored; the result set is generated dynamically when the view is accessed.	Query results are stored physically in the database, consuming additional storage space.
Performance	Slower for complex queries since the result set is computed dynamically on each access.	Faster as results are precomputed and stored, reducing computation time.
Update Behavior	Automatically reflects changes in the underlying tables since data is retrieved dynamically.	Needs manual or automatic refresh to update the stored data when underlying tables change.
Storage Cost	No additional storage cost since data is not physically stored.	Requires extra storage as it saves query results.
Maintenance Cost	No maintenance cost, as views are dynamically updated with no stored data.	Involves maintenance cost due to periodic refreshes to keep data synchronized with base tables.
SQL Standards	Fully standardized and supported by all major database systems.	Not fully standardized; support and implementation vary across database systems.
Use Cases	Best for scenarios where data is accessed infrequently and requires up-to-date values.	Ideal for frequently accessed data where performance is critical, such as reporting and analytics.

The background of the slide is an aerial photograph of a dry, cracked landscape, possibly a salt flat or a desert. The image is overlaid with a semi-transparent blue filter. In the center, there is a white rectangular box containing the section title.

III - Entretien d'une database

Entretien d'une database - Pourquoi ?

Dans une entreprise, pas tout le monde n'a un accès direct aux données, et encore moins à la modifications

- Pas forcément Expert en Data
- Trop occupé pour se plonger dans les données
- ...

⇒ Demande du Business pour des données

⇒ Alimentation de la database par le Business (Besoin de Garde-fou)

⇒ Nettoyage / Normalisation des données

⇒ Besoins de données de qualité

Entretien d'une database - Contraintes

Pour s'assurer la qualité des données, il est possible de mettre des contraintes :

- NOT NULL - Ensures that a column cannot have a NULL value
- UNIQUE - Ensures that all values in a column are different
- PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- FOREIGN KEY - Prevents actions that would destroy links between tables
- CHECK - Ensures that the values in a column satisfies a specific condition
- DEFAULT - Sets a default value for a column if no value is specified
- CREATE INDEX - Used to create and retrieve data from the database very quickly

Entretien d'une database - Access?

Dans une entreprise, pas tout le monde n'a un accès direct aux données, et encore moins à la modifications

Qui a le droit de toucher au données ? Comment les différencier ?

CRUD : Create Read Update Delete

Rédaction de spécifications fonctionnelles

Entretien d'une database - Gestion des droits

Rôle	Description
db_owner	Les membres du rôle de base de données fixe db_owner peuvent effectuer toutes les activités de configuration et de maintenance sur la base de données et peuvent également DROP la base de données dans SQL Server. (Dans SQL Database et Azure Synapse, certaines activités de maintenance nécessitent des autorisations au niveau du serveur et ne peuvent pas être effectuées par le rôle db_owners.)
db_securityadmin	Les membres du rôle de base de données fixe db_securityadmin peuvent modifier l'appartenance au rôle pour les rôles personnalisés uniquement et gérer les autorisations. Les membres de ce rôle peuvent potentiellement élever leurs privilèges et leurs actions doivent être supervisées.
db_accessadmin	Les membres du rôle de base de données fixe db_accessadmin peuvent ajouter ou supprimer l'accès à la base de données des connexions Windows, des groupes Windows et des comptes de connexion SQL Server.

Rôle	Description
db_backupoperator	Les membres du rôle de base de données fixe db_backupoperator peuvent sauvegarder la base de données.
db_ddladmin	Les membres du rôle de base de données fixe db_ddladmin peuvent exécuter n'importe quelle commande DDL (Data Definition Language) dans une base de données. Les membres de ce rôle peuvent potentiellement élever leurs privilèges en manipulant du code qui peut être exécuté avec des privilèges élevés et leurs actions doivent être surveillées.
db_datawriter	Les membres du rôle de base de données fixe db_datawriter peuvent ajouter, supprimer et modifier des données dans toutes les tables utilisateur. Dans la plupart des cas d'usage, ce rôle est combiné avec db_datareader appartenance pour permettre la lecture des données à modifier.
db_datareader	Les membres du rôle de base de données fixe db_datareader peuvent lire toutes les données de toutes les tables et vues utilisateur. Les objets utilisateur peuvent exister dans n'importe quel schéma, sauf <code>sys</code> et <code>INFORMATION_SCHEMA</code> .
db_denydatawriter	Les membres du rôle de base de données fixe db_denydatawriter ne peuvent ajouter, modifier ou supprimer aucune donnée des tables utilisateur d'une base de données.

Database cassée

Dans le cas où une personne mal intentionnée efface/supprime la database, que faire?

Pour éviter cela : **Backup**

Aussi, selon le type de projet, avoir une historisation des données peut être intéressant

Comment mettre un backup automatique :

<https://www.digitalocean.com/community/tutorials/how-to-schedule-automatic-backups-for-postgresql-with-pgagent-in-pgadmin>